

# Using Modular Self-reconfiguring Robots for Locomotion

Keith Kotay, Daniela Rus, Marsette Vona  
Department of Computer Science  
Dartmouth  
Hanover NH 03755, USA  
{rus,kotay,mav}@cs.dartmouth.edu

**Abstract:** We discuss the applications of modular self-reconfigurable robots to navigation. We show that greedy algorithms are complete for motion planning over a class of modular reconfigurable robots. We illustrate the application of this result on two self-reconfigurable robot systems we designed and built in our lab: the *robotic molecule* and the *atom*. We describe the modules and our locomotion experiments.

## 1. Introduction

Self-reconfiguring robots have the ability to adapt to the operating environment and the required functionality by changing shape. They consist of a set of identical robotic modules that can autonomously and dynamically change their aggregate geometric structure to suit different locomotion, manipulation, and sensing tasks. A primary design goal for a self-reconfiguring robot is to allow the robot to assume any geometric shape. For example, a self-reconfiguring robot system could self-organize as a snake shape to pass through a narrow tunnel and reorganize as a multi-legged walker upon exit to traverse rough terrain. Self-reconfiguring robots are suited for a range of applications that require the geometric modification of a part and are characterized by incomplete a priori task knowledge. Such a robot could match its geometric structure to the shape of the surrounding terrain for versatile locomotion. This can be achieved by requiring the robot to metamorphose from one shape to another to best match the shape of the terrain in a statically stable gait, as illustrated in Figure 1.



Figure 1. This figure demonstrates using shape metamorphosis for locomotion. A statically stable gait is used to translate the robot from left to right.

In our previous work [7, 5, 17, 18, 19] we describe two different robot systems capable of self-reconfiguration: the *Robotic Molecule* system and the *Robotic Crystal* system. In this paper we examine using self-reconfiguration for locomotion and we describe our experimental results in simulation and on the hardware units we built in our lab. These results have the flavor of [8], where

we examine locomotion with Inchworm robots.

## 2. Two Self-reconfigurable Robot Systems

### 2.1. The Molecule

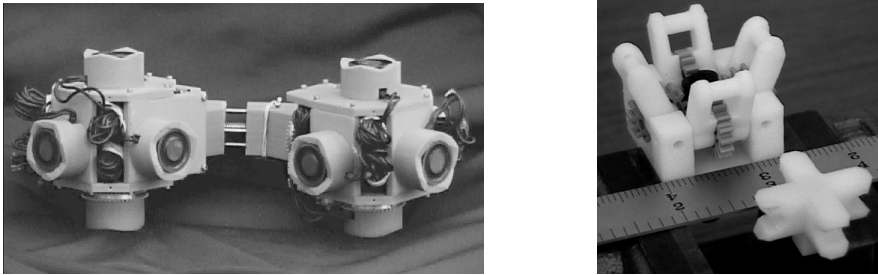


Figure 2. **(Left)** The robotic Molecule. The Molecule is composed of two atoms, connected by an right-angle rigid bond. The Molecule has 4 degrees of freedom: two rotational degrees of freedom about the bond and one rotational degree of freedom per atom about a single inter-Molecule connector. The connectors have been implemented with electromagnets. **(Right)** The prototype gripper connection mechanism. The gripper is a male-female design. The male component is in the upper left and the female component is in the lower right. Molecules will either have all male components or all female components as connectors. This does not cause a problem because the Molecule design naturally partitions 3D space into two regions. A single Molecule can only occupy one of the regions and can only connect to Molecules in the other region.

A Molecule robot [6] consists of multiple units called *Molecules*; each Molecule consists of two *atoms* linked by a rigid connection called a *bond* (see Figure 2). Each atom has five inter-Molecule connection points and two degrees of freedom. One degree of freedom allows the atom to rotate 180 degrees relative to its bond connection, and the other degree of freedom allows the atom (thus the entire Molecule) to rotate relative 180 degrees relative to one of the inter-Molecule connectors at a right angle to the bond connection. We have already prototyped the Molecule (see Figure 2.)

Our current design uses R/C servomotors for the rotational degrees of freedom. A new feature of our prototype is the use of a gripper-type connection mechanism (see Figure 2). In our previous design we used electromagnets as the connection mechanism, but electromagnets have several disadvantages including continuous power consumption to maintain connections and requiring a sheath to prevent unwanted rotation about the axis of connection. Since a sheath must extend beyond the bounding sphere of the atom to allow it to interlock with its mating sheath, a binding condition in introduced restricting mating motion to a face-to-face approach (a sliding approach, in which the two mating faces come into contact by sliding past each other is not possible because of sheath collisions). A gripper-type connection mechanism, in which the

gripper arms can retract into the bounding sphere of the atom allows sliding face-to-face approaches and atom rotations in place. Also, since the gripper arms are driven by a non-backdrivable worm gear mechanism, they will maintain their grip when electrical power is no longer applied, decreasing the power consumption of Molecule self-reconfiguration.

The rotating connection points on each atom are the only connection points required for Molecule motion. The other connection points are used for attachment to other Molecules to create stable 3D structures. Each Molecule also contains a microprocessor and the circuitry needed to control the servomotors and connectors. The diameter of each atom is 4 inches (10.2 cm.), making the atom-atom distance in the Molecule approximately 5.7 inches (14.4 cm.). The weight of the Molecule is 3 pounds (1.4 kg.).

## 2.2. Molecule Motion

An individual Molecule has the following basic motion capabilities: (1) linear motion in a plane on top of a lattice of identical Molecules, irrespective of the absolute orientation of the plane; (2) convex 90-degree transitions between two planar surfaces composed of Molecules; and (3) concave 90-degree transitions between two planar surfaces composed of Molecules.

The details of controlling these motions using the 4 molecular DOFs are provided in [6]. Figure 3 illustrates the linear walk algorithm.

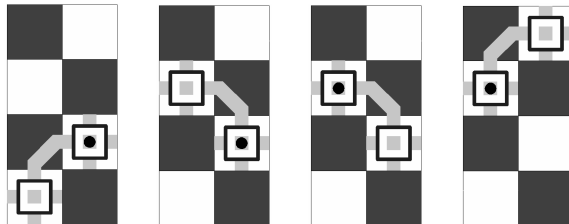


Figure 3. A linear walk sequence. The checkered surface represents a plane of Molecules. An atom with a black dot is attached to the Molecule below it. The left image represents the initial configuration. A clockwise rotation of 90 degrees about the connected atom produces the next image. The atoms then swap attachment as indicated by the movement of the black dot. Finally, a counterclockwise rotation of 90 degrees produces the right image. Another attachment swap would return the Molecule to its initial pose, translated by two squares in the vertical direction. A similar sequence could be used to translate the Molecule horizontally. Thus, these sequences of moves are sufficient for Molecular translation to any pair of white squares in the plane.

## 2.3. The Atom

The Crystalline robot consists of a set of modules called *Atoms*; each Atom is a mechanism that has some of the motive properties of muscles, that can be closely packed in 3D space, and that can attach itself to similar units. We chose a design based on cubes with connectors to other modules in the middle of each face. The idea is to build a cube that can contract by a factor of two and expand

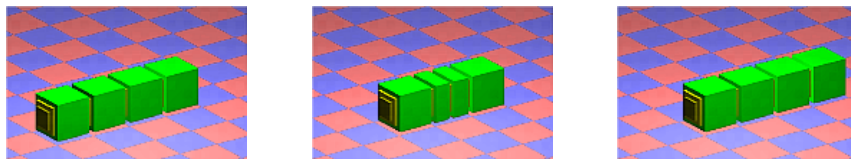


Figure 4. Three snapshots from a simulation of locomotion using Crystalline robots. The left image shows the initial state. The middle image shows the robot after shrinking two modules in the direction of motion. The right image shows the robot after relaxing the shrunk modules in the direction of motion. Notice that the entire structure moved forward one unit, in an inchworm-like fashion.

to the original size (see Figure 4). We wish to effect compression along all three principal directions (e.g.,  $x, y, z$ ) individually or in parallel. We call the module an *Atom*, and each connector a *bond*. Figure 5 shows a design for the mechanics of a two-dimensional (square rather than cubic) implementation of the Atom and Figure 6 shows the physical prototype. We use complimentary rack and pinion mechanisms to implement the contraction and expansion actuation for the two-dimensional prototype. The connection mechanisms are based on a *channel and key* concept. When fully contracted, the Atom is a square with a 2 inch side. When fully expanded, the Atom is a square with a 4 inch side. The height of the Atom is 7 inches and its weight is 12 ounces.

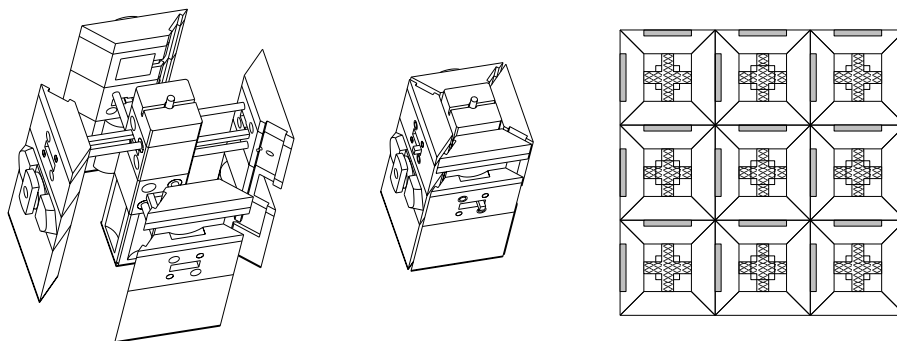


Figure 5. The mechanics of a 2D Atom actuated by complimentary rack-and-pinion mechanisms. The Atom is 4 inches tall (not including electronics, which are not shown). When expanded (left), the Atom occupies a 4 inch square; when contracted (middle) the Atom occupies a 2 inch square. The right figure shows a tiling of nine compressed 3DOF Atoms. Note that every inter-Atomic interface contains exactly one active connection mechanism.

The two-dimensional version of the Crystalline Atomic module (see Figure 6) was created based on the CAD designs shown in Figure 5 (left and middle). The module has an expansion/contraction ration of 2. All the faces of the Atom have to be fully extended or fully contracted. Each face of the

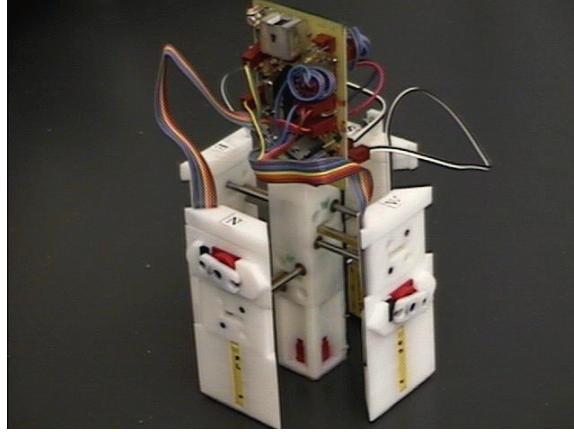


Figure 6. The physical prototype for the Crystalline Atom.

module has a connection slot. However, only two out of the four faces<sup>1</sup> have active connection slots (see Figure 5 (right)). These active slots provide a key-and-lock mechanism for forming rigid connections with adjacent modules. Thus, the entire unit can be realized with three degrees of freedom: one to expand/contract the faces of the Atom, and two for the active connectors. All three degrees of freedom can be implemented with binary actuators. Since Atoms can never rotate relative to one another, the use of two rather than four connectivity degrees of freedom leads to no mechanical limitations. Every inter-Atomic interface of a structure will have one active connection mechanism. The module has on-board electronics and four 3V 2/3A size Lithium batteries, so that it can function untethered.

A Crystalline Atom can connect with identical modules to create Crystalline robot systems. Only lattices whose faces are normal to the  $x$ ,  $y$ , and  $z$  axes can be created using Crystalline robots. By manipulating the size of the Atom, it is possible to approximate any finite solid shape to an arbitrary precision using Crystalline modules<sup>2</sup>.

Each Atom contains an on-board processor (Amtel AT89C2051 microcontroller), power supply (four 2/3 A Lithium batteries), and support circuitry, which allows both fully untethered and tethered operations. Atoms are connected by a wired serial link to a host computer to download programs. For untethered operations, an experiment specific operating program specified as a state sequence is first downloaded over a tether. When the tether is removed, an on-board IR receiver is used to detect synchronization beacons from the host.

---

<sup>1</sup>The active connection slots are situated on adjacent faces, which allows any lattice of Crystalline Atoms to be fully connected.

<sup>2</sup>The aliasing error for any shape on a raster display can be arbitrarily reduced by increasing the resolution of the display.

### 3. Locomotion with Modular Self-reconfigurable Robots

In this section we examine locomotion capabilities for modular self-reconfiguring robot systems.

Self-reconfigurable robots consist of modules that can move relative to each other. The modules can climb on top of each other<sup>3</sup>, slide relative to each other, etc. This enables the modules of the robot to move outside the plane supporting the robot. Thus, a modular self-reconfigurable robot is capable of climbing on top of obstacles. In this section we examine the power of this capability to motion planning in the absence of maps. For example, we may consider a factory floor where dynamic obstacles make it impossible to supply the robot with an accurate map. The model for this problem is a self-reconfiguring robot that starts at a known location and is to find its way to a goal location, identifiable by a beacon.

In this section we do not assume any specific design for the unit module. To preserve generality, we assume a unit-modular system where an individual module can move linearly and make convex and concave transitions relative to a collection of modules. These capabilities enable an entire class of robots to move linearly and make convex and concave transitions in the environment<sup>4</sup>.

A simple strategy, such as the right hand rule, or the on-line navigation algorithms proposed by [9], may be employed to find a path to the goal. However, because modular self-reconfiguring robots have the ability to move out of plane, we propose a simpler algorithm for this problem. The basic idea is to move the robot greedily in the direction of the goal. When an obstacle is reached, instead of going around the obstacle, which is the technique employed by robots confined to move in the plane (for example wheel-based robots), self-reconfiguring robots can simply climb over the obstacles, maintaining their original heading. The algorithm described in Figure 7 summarizes this intuition.

```
(define (greedy-navigation start goal gravitation-direction)
  (align-robot (make-path start goal))
  (loop (cond ((at-goal?)
              'stop)
             ((obstacle? 'front-IR)
              (concave-transition))
             ((free-space? 'front-foot-IR)
              (convex-transition))
             (else (linear-step))))))
```

Figure 7. A greedy algorithm for on-line navigation. The robot moves in the direction of the goal using `step`. If an obstacle is encountered, the robot uses `convex-transition` and `concave-transition` to climb over the obstacle.

**Theorem 1** *Suppose a self-reconfiguring robot has to travel from an initial*

---

<sup>3</sup>Each system implements this operation in a different way, using its own specific actuation capabilities.

<sup>4</sup>We have shown these capabilities for Molecule robots in [7] and for Atoms in [19]. The Experiments section details how the two systems accomplish such motions.

location  $S$  to a goal location  $G$  in an unknown environment with piecewise-planar segments. The greedy algorithm in Figure 7 is complete and takes  $O(1)$  time to compute a path to the goal, provided each segment is wide enough to allow the robot to step on it.

**Proof:**

The path followed by the self-reconfiguring is obtained by intersecting the environment with the plane defined by  $S$ ,  $G$ , and the direction of gravitation. This plane can be computed in  $O(1)$  time. Starting in this plane, the robot will move greedily towards the goal. At each step, the robot will use its heading and sense the direction of gravitation<sup>5</sup> to ensure that its motion stays confined to the motion plane.

This greedy algorithm operates as hill climbing towards the goal and it is complete. The actual path is a simple polygon that connects the  $S$  to  $G$ ; thus the Inchworm is guaranteed to reach  $G$ . The total length of the path is of length at most  $2 \sum H + D$ , where  $D$  is the straight line distance from  $S$  to  $G$  and  $\sum H$  sums the heights of the obstacles in the space.

Note that this on-line algorithm (see Figure 7) allows the robot to reach the goal provided the robot can place itself completely on each edge of the path. This condition translates into the assumption that all the polygonal edges of its path are of length at least  $k + 1$  (where  $k$  is the size of one module) for robots with discrete orientations for their modules such as ours.  $\square$

This on-line motion planning algorithm will not always move the robot on the shortest path to the goal. For example, if the environment has very high but skinny obstacles, the robot will do extra work to reach the goal. The advantage of on-line navigation with self-reconfiguring robots relies on the capability of such robots to move out of plane. The resulting algorithm is very simple and it only requires computing the direction of motion for each step. This algorithm is significantly simpler than strategies such as [9] where the robot has to completely surround an obstacle to compute the best way to move.

## 4. Experiments

### 4.1. Experiments with Molecules

We have constructed two prototype modules and used them to perform experiments to evaluate the feasibility of implementing locomotion using self-reconfiguration. Two other modules are currently under construction. We have also developed a simulator that allowed us to experiment with locomotion algorithms applied to systems consisting of many more modules.

A two-molecule system can walk on a floor of connectors. A four-Molecule Robot can use the individual Molecule actuation to translate and rotate in the plane without needing any additional support from external connectors [7]. The four-Molecule linear translation can be extended to a  $2k$ -Molecule linear chain of Molecule pairs. An eight-Molecule system can climb stairs. Several

---

<sup>5</sup>This part requires a sensor such as a potentiometer to sense the direction of gravitation.

algorithms can implement this motion. One possible algorithm to implement this task is a *Slinky*-style motion: the modules get organized into a tower on the plane of one step. The tower is then reversed from the top on the next step. How do we insure that all the modules of the tower (including those at the base, that provide support for the tower) get moved to the next step? Figure 8 illustrates this issue using robots that consist of *Robotic Molecules*. To remove the bottom modules from the lower step the entire structure needs to be supported and balanced by the robotic molecules on the upper step. This poses constraints on the number of modules that need to be located on the upper step prior to breaking ground contact on the bottom step.

Self-reconfiguring robots can adapt to the geometry of the terrain, but planning for such versatility requires additional constraints about the dynamics of the robot structure—our algorithm for stair climbing guarantees dynamic stability during motion.

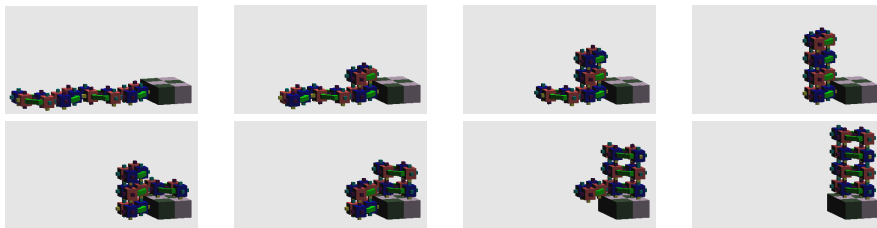


Figure 8. A stair climbing sequence using Robotic Molecules [7]. The light and dark gray cubes on the right of each image represents the first step in a staircase.

We are implementing the four molecule walk on a plane. We have already experimented with a two-molecule system that can successfully climb on top of each other, thus implementing the basic rolling primitive necessary for stepping forward. This two-molecule system requires the support of connectors in the environment.

#### 4.2. Experiments with Atoms

We have constructed ten prototype modules and used them to perform experiments to evaluate the feasibility of using multiple Atoms to demonstrate reconfiguration.

To facilitate experimentation, a row of 8 fixed passive connectors was constructed to simulate the surface of a Crystal. This arrangement not only frees us from having to construct many units at the outset, but it also allows us to perform experiments that are focused narrowly on the specific activities under study. The fixed connectors are placed as they would be for a flat Crystal surface composed of 8 contracted Atoms. In the descriptions that follow, we will refer to two of the prototype Atoms as **a** and **b**, and we will number the fixed connectors **0–7**. The North and West faces of **a** and **b** (those that contain active connection mechanisms) will be referred to as **a.n/b.n** and **a.w/b.w**, respectively, and the South and East faces will be similarly named. **a** and **b**

are always oriented so that **a.n** and **b.n** are facing the row of fixed connectors.

Our locomotion experiment was designed to evaluate whether Atoms could work together to effect a reconfiguration. Initially, both **a** and **b** were contracted. **a** was connected to **0** (at **a.n**) and **b** was connected to **1** (at **b.n**). **a** and **b** were connected together at **b.w**. The Atoms were programmed with state sequences designed to perform a variant of inchworm translation along the fixed connectors:

- |  |
|--|
| <ol style="list-style-type: none"> <li>1. free <b>b.n</b> from 1</li> <li>2. expand <b>a</b></li> <li>3. expand <b>b</b></li> <li>4. connect <b>b.n</b> to 2</li> <li>5. disconnect <b>a.n</b> from 0</li> <li>6. contract <b>a</b> and <b>b</b></li> <li>7. connect <b>a.n</b> to 1</li> <li>8. repeat</li> </ol> |
|--|

This sequence is illustrated in Figure 9, and Figure 10 presents several photographs of the Atom prototype hardware performing the experiment.

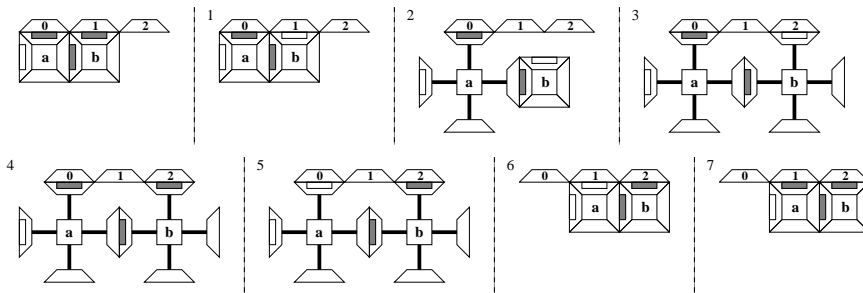


Figure 9. The second experiment tests an inchworm propagation algorithm.

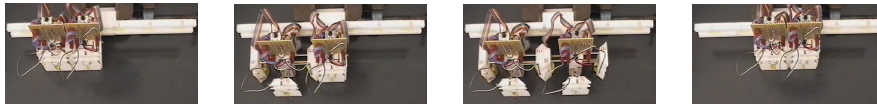


Figure 10. Several snapshots of the Atom prototype hardware performing the inchworm experiment.

## 5. Related Work

We are inspired by pioneering work in self-reconfiguring robotics. In [2], Fukuda et al propose a cellular robotic system to coordinate a set of specialized modules. Several specialized modules and ways of composing them were proposed. In [21] Yim studies multiple modes of locomotion that are achieved physically

by manually composing a few basic elements in different ways. This work also presents extensive examples of locomotion and self-reconfiguration in simulation. In [10, 23, 20, 11], Murata et al consider a system of modules that can achieve planar motion by walking over one another. The reconfiguration motion is actuated by varying the polarity of electromagnets that are embedded in each module. More recently [12] this group developed a twelve DOF module capable of three-dimensional motion. In [13] Chirikjian et al describe metamorphic robots that can aggregate as two-dimensional structures with varying geometry. The modules are deformable hexagons. This work also examines theoretical bounds for planning the self-reconfiguring motion of such modules.

## 6. Discussion

We have discussed how modular self-reconfiguring robots can be used to implement versatile locomotion. The ability of such robots to move out of the plane supporting the robot enables them to climb on top of obstacles. Thus, greedy algorithms that move the robot on a straight line to the goal (which might involve climbing over obstacles) are complete for a class of environments where the size of the obstacles is compatible with the size of the robots' discrete steps. We have illustrated our point on two very different robot systems we have designed and built in our lab. We have reported on our experiments with these robots.

## Acknowledgements

This paper describes research done in the Dartmouth Robotics Laboratory. Support for this work was provided through the NSF CAREER award IRI-9624286, NSF award IRI-9714332, NSF award EIA-9901589, NSF award IIS-98-18299, and NSF IIS 9912193

## References

- [1] G. Chirikjian, A. Pamecha, and I. Ebert-Uphoff. Evaluating Efficiency of Self-Reconfiguration in a Class of Modular Robots. *Journal of Robotic Systems*, Vol. 13., No. 5., May 1996.
- [2] T. Fukuda and Y. Kawauchi. Cellular robotic system (CEBOT) as one of the realization of self-organizing intelligent universal manipulator. In *Proceedings of the IEEE Conference on Robotics and Automation*, pp. 662-667, 1990.
- [3] G. Hamlin and A. Sanderson. Tetrabot modular robotics: prototype and experiments. In *Proceedings of the IEEE/RSJ International Symposium of Robotics Research*, pp 390-395, Osaka, Japan, 1996.
- [4] Kazuo Hosokawa, Isao Shimoyama, and Hirofumi Miura. Dynamics of self-assembling systems — analogy with chemical kinetics. *Artificial Life*, 1(4), 1995.
- [5] K. Kotay, D. Rus, M. Vona, and C. McGray. The self-reconfigurable robotic molecule. In *Proceedings of the 1998 International Conference on Robotics and Automation*, 1998.
- [6] K. Kotay and D. Rus. Motion Synthesis for the Self-reconfiguring Robotic Molecule. In *Proceedings of the 1998 International Conference on Intelligent Robots and Systems*, 1998.

- [7] K. Kotay and D. Rus. Locomotion Versatility through Self-reconfiguration In *Robotics and Autonomous Systems*, 26 (1999), pp 217–232.
- [8] K. Kotay and D. Rus. The Inchworm Robot: A Multi-functional System In *Autonomous Robots*, Vol. 8, no. 2, pp 53–69, 2000.
- [9] V. Lumelsky. Algorithmic issues of sensor-based robot motion planning. in *Proceedings of the 26th Conference on Decision and Control*, pp 1796–1801, Los Angeles, CA 1987.
- [10] S. Murata, H. Kurokawa, and Shigeru Kokaji. Self-assembling machine. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, San Diego, 1994.
- [11] S. Murata, H. Kurokawa, K. Tomita, and Shigeru Kokaji. Self-assembling method for mechanical structure. In *Artif. Life Robotics*, 1:111–115, 1997.
- [12] S. Murata, H. Kurokawa, E. Yoshida, K. Tomita, and S. Kokaji. A 3-D Self-Reconfigurable Structure. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, Leuven, 1998.
- [13] A. Pamecha, C-J. Chiang, D. Stein, and G. Chirikjian. Design and implementation of metamorphic robots. In *Proceedings of the 1996 ASME Design Engineering Technical Conference and Computers in Engineering Conference*, Irvine, CA 1996.
- [14] A. Pamecha, I. Ebert-Uphoff, and G. Chirikjian. Useful Metrics for Modular Robot Motion Planning. *IEEE Transactions on Robotics and Automation* pp531-545, Vol.13, No.4, August 1997.
- [15] C. Paredis and P. Khosla. Design of Modular Fault Tolerant Manipulators. In *The First Workshop on the Algorithmic Foundations of Robotics*, eds. K. Goldberg, D. Halperin, J.-C. Latombe, and R. Wilson, pp 371-383, 19 95.
- [16] D. Rus. Self-Reconfiguring Robots. *IEEE Intelligent Systems*, 13(4), 2-5, July/August 1998
- [17] D. Rus and M. Vona. Self-Reconfiguration Planning with Unit Compressible Modules. *Proceedings of the 1999 IEEE International Conference on Robotics and Automation*, pp 2513–2520, Detroit, MI, 1999.
- [18] D. Rus and M. Vona. A Physical Implementation of the Crystalline Robot. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, CA, 2000.
- [19] D. Rus and M. Vona. Crystalline Robots: Self-reconfiguration with Unit-compressible Modules. *Autonomous Robots* Vol. 10, no. 1, pp 107–124, 2001.
- [20] K. Tomita, S. Murata, E. Yoshida, H. Kurokawa, and S. Kokaji. Reconfiguration method for a distributed mechanical system. In *Distributed Autonomous Robotic Systems 2*, pp 17–25, Springer Verlag 1996.
- [21] M. Yim. A reconfigurable modular robot with multiple modes of locomotion. In *Proceedings of the 1993 JSME Conference on Advanced Mechatronics*, Tokyo, Japan 1993.
- [22] M. Yim. Polypod II. <http://www.parc.xerox.com/spl/members/yim/>
- [23] E. Yoshida, S. Murata, K. Tomita, H. Kurokawa, and S. Kokaji. Distributed Formation Control of a Modular Mechanical System. In *Proceedings of the 1997 International Conference on Intelligent Robots and Systems*, 1997.